

1. 序論

大容量の記憶デバイスの登場や、ネットワークの高速化により、大量のデータを高速に処理することが求められている。その問題の解決のために、一つの処理に対して、ネットワーク上で接続した複数の計算機で対応する、仮想並列計算機というシステムが考えられる。本研究では、MPI(Message Passing Interface)¹⁾を用いて、その有用性を検証する。

2. 研究内容

2.1 研究目的

1 台の計算機を用いて処理した場合と比べて、MPIを用いて複数の計算機で並列処理した場合にどの程度処理時間を短縮できるかを計測し、並列計算での処理の有用性を検証するのが本研究の目的である。

2.2 研究方法

本研究では計算機 4 台を用いて MPI 環境を構築する。MPI の有用性を検証するための問題として、最小全域木問題を用いる。最小全域木問題とは、重み付無向グラフ G が与えられたとき、 G の閉路を含まない連結な部分グラフの中で、辺の重みの総和が最小のものを求める問題である。

本研究では、頂点数が 10,20,30,40 の重み付無向グラフに対し、その最小全域木をそれぞれ 1,2,3,4 台の計算機で計算し、その実行時間を測定する。本研究で作成した最小全域木問題の計算するプログラムは Sollin³⁾ のアルゴリズムを元に行っている。Sollin のアルゴリズムを以下に示す。

1. 各頂点 v において、 v に接続する辺のうち、最も重みの軽い辺 (v,u) を選択し、 u を v の親とする。
2. 各頂点 v において、 v の根 $r[v]$ を求める。
3. 各頂点 v に接続されている辺と隣接する頂点の情報を $r[v]$ に送り、同一の根を持つ頂点の集合を 1 つの頂点としてグラフを構成しなおす。

上記の 1,2,3 の操作を $\log n$ 回繰り返す。

また、本研究では上記のアルゴリズムの実装に MPICH2²⁾ を用いた。MPICH2 とは、MPI の実装ライブラリであり、rsh/ssh 越しに通信を行うものである。

3. 結果・考察

表 1 に頂点数 10,20,30,40 の重み付無向グラフに対して、その全域木をそれぞれ 1,2,3,4 台の計算機を用いて求めた場合のプログラムの内部計算にかかった時間を示し、表 2 に

表 1 内部計算時間と計算機数の関係

頂点数 \ 台数	1	2	3	4
10	0.000010	0.000008	0.000006	0.000005
20	0.000017	0.000012	0.000009	0.000009
30	0.000027	0.000022	0.000015	0.000014
40	0.000038	0.000031	0.000024	0.000021

(m 秒)

表 2 全体の処理時間と計算機数の関係

頂点数 \ 台数	1	2	3	4
10	0.0045	0.012	0.016	1.3
20	0.006	0.013	0.023	2.0
30	0.012	0.018	0.0055	3.0
40	0.023	0.039	0.064	3.2

(m 秒)

各計算機間の通信時間を含めた全体の処理時間を示す。

表 1 より、計算機の台数を増やすことにより、内部計算の時間は短縮されることが示されたが、表 2 により、プログラムの全体の処理時間は計算機の台数が増える程、処理にかかる時間は長くなっていることが示される。これは、各計算機間でのデータの通信に時間がかかっているためだと考えられる。

4. 結論

本研究では、並列計算アルゴリズムの有用性を検証するために MPI を用いて最小全域木問題を解く時間を計測した。内部の計算時間は短縮することができたが、全体の処理時間は台数が増えるごとに時間がかかるようになった。これは、各計算機間での通信に時間がかかったためであり、MPI の有用性を示すためには、よりよい通信の環境を構築することが考えられる。

参考文献

- 1) P.Pacheco 著, 秋葉博訳, MPI 並列プログラム, 培風館 (2001)
- 2) MPICH2, <http://www.mcs.anl.gov/research/projects/mpich2>, Argonne national laboratory.
- 3) J.JáJá 著, An Introduction to Parallel Algorithms, Addison-Wesley Professional (1992).